

Claims

1 1. In a computer system in which a first thread and a
2 second thread execute concurrently in a common address
3 space, a method of facilitating the handling of an external
4 event by said first thread, comprising the steps of:

5 sending a quiesce event from said first thread to said
6 second thread to cause said second thread to quiesce;

7 suspending execution of said first thread until said
8 second thread has quiesced in response to the quiesce event
9 sent to that thread; and

10 resuming execution of said first thread to process said
11 event when said second thread has quiesced in response to
12 the quiesce event sent to that thread.

1 2. The method of Claim 1 wherein said second thread is one
2 of a plurality of additional threads executing concurrently
3 with said first thread in said address space, said quiesce
4 event being sent from said first thread to each of said
5 additional threads.

1 3. The method of Claim 2 wherein execution of said first
2 thread is suspended until each of said additional threads
3 has quiesced in response to the quiesce event sent to that
4 thread.

1 4. The method of Claim 3 wherein the last of said
2 additional threads to quiesce resumes execution of said
3 first thread.

1 5. The method of Claim 1 wherein said quiesce event is
2 sent from said first thread to said second thread in
3 response to the detection of said external event by said
4 first thread.

1 6. The method of Claim 1 wherein said quiesce event is a

2 termination event causing said second thread to terminate.

1 7. The method of Claim 1 wherein said quiesce event is a
2 suspension event causing said second thread to suspend.

1 8. The method of Claim 1 wherein said step of sending a
2 quiesce event from said one thread to said second thread
3 comprises the step of interrupting the execution of said
4 second thread to give control to a quiesce exit routine.

1 9. The method of Claim 8 wherein said quiesce exit routine
2 checks to determine whether said second thread is holding
3 any critical resource and quiesces said second thread only
4 if it determines that the second thread is not holding any
5 critical resource.

1 10. The method of Claim 1, comprising the further steps of:
2 determining whether said second thread is holding any
3 critical resource; and
4 quiescing said second thread only if it is determined that
5 the second thread is not holding any critical resource.

1 11. The method of Claim 10, further comprising the step of
2 releasing any critical resource held by said second thread
3 before quiescing said second thread.

1 12. In a computer system in which a first thread and a
2 second thread execute concurrently in a common address
3 space, a method of facilitating the handling of an external
4 event by said first thread, comprising the steps of:
5 sending a suspension event from said first thread to
6 said second thread to cause said second thread to suspend;
7 suspending execution of said first thread until said
8 second thread has suspended in response to the suspension
9 event sent to that thread;
10 resuming execution of said first thread to process said

11 event when said second thread has suspended in response to
12 the suspension event sent to that thread; and
13 resuming said second thread following the processing of
14 said event by said first thread.

1 13. The method of Claim 12 wherein a plurality of
2 additional threads execute concurrently with said first
3 thread in said address space, said quiesce event being sent
4 from said first thread to each of said additional threads.

1 14. The method of Claim 13 wherein execution of said first
2 thread is suspended is suspended until each of said
3 additional threads has quiesced in response to the quiesce
4 event sent to that thread.